Diese Website verwendet Cookies von Google, um Dienste anzubieten und den Traffic zu analysieren. Deine IP-Adresse und dein User-Agent werden zusammen mit Messwerten zur Leistung und Sicherheit für Google freigegeben. So können Nutzungsstatistiken generiert, Missbrauchsfälle erkannt und behoben und die Qualität des Dienstes gewährleistet werden.

WEITERE INFORMATIONEN OK

Sunday, September 02, 2007

Bilinear group cryptography

An important recent development in public key cryptography is the bilinear group, which for abstract algebra wonks is defined as follows (if you're not into abstract algebra feel free to skip to below):

> Bilinear groups are a set of three abstract algebraic groups, G1, G2 and GT , together with a deterministic function e, called a bilinear map, that takes as input one element from G1 and one element from G2 and outputs an element in GT . Suppose all three groups have order Q, element g1 generates group G1, and element g2 generates group G2. Then, one special property called bilinearity of the map e is that for all a, b < Q, we have that e(g1^a , g2^b) = e(g1, g2)^ab. This new element, e(g1, g2)^ab, is in group GT . The key observation is what happens to the exponents, a and b, during the mapping: they are multiplied. The group GT is distinct from both G1 or G2; thus the output of the mapping cannot be fed back into the map e as input.

Elliptic curves are generally used for the groups, although bilinear schemes in at least some other algebras are also possible.

Two of the main applications of bilinear groups are proxy re-signatures and proxy re-encryption. In proxy re-signatures, a semi-trusted party transforms Alice's public key signature into Bob's. The proxy does not have, cannot derive, and thus cannot sign with either Bob's secret key or with Alice's, but can only transform Alice's signature into Bob's. The proxy re-signer is thus "semi-trusted" -- it is trusted with some things we normally would trust a proxy signer with, but not with others. For example it is not trusted with either Alice's or Bob's private key, only with a special key that allows the signature transformation.

The target signature could also be a group signature. Thus, for example, Alice could sign her e-mail with her own digital signature, and a proxy re-signer sitting on the corporate e-mail firewall could re-sign the e-mail with the corporate group signature.

Proxy re-signers can be chained in a series, so that signature A is transformed by proxy AB into signature B, which is transformed by proxy BC into signature C, and so on. The last signature Z proves that the message was signed by each proxy in the chain in order. Proxy re-signers can also be chained together in a tree or directed acyclic graph. (Note that threshold signatures by contrast do not require or prove that the signatures took place in a particular order).

Proxy re-encryption is the same idea for public key encryption, with the added bonus that the re-encryptor can't read the message. So, for example, we could have the following scheme to restrict the distribution of content:

Pages

- Home

About Me

Nick Szabo

"A premier thinker about history, law and economics, and the lessons they have for security." -- Adam Shostack, Emergent Chaos

"Szabo comes out with these essays that leave me in awe." -- Brian Dunbar

"Reading material that is eclectic, challenging and endlessly fascinating." -- Sean McGrath, Propylon

"Like most blogs worth my attention, this blog is updated only infrequently. That is because the authors of blogs worth my attention only post when they have something to say that is true, relevant and not already known by their audience. Most of the human race does not have the skill to know when an idea has these three properties. The skill is particularly rare in the fields of politics and economics, which is why this blog is such a rare and valuable thing." -- Richard Hollerith

View my complete profile

(1) Content owner Alice encrypts her content with her public key and publishes it to proxies P1, P2, etc., along with re-encryption keys AC1, AC2, etc. for each customer.

(2) Proxy allows customer to access the content only if paid. When paid, the proxy re-encrypts to the customer using the re-encryption key for that customer.

The proxies themselves are trusted neither with an ability to view the content nor with the discretion to distribute to additional customers not desired by content owner Alice. The proxy is trusted only to restrict access to a customer. (I present this scheme mainly just to illustrate what proxy re-encryption does. As an application, this particular content distribution scheme seems to me to only be useful if it somehow lowers transaction costs to route all payments through proxies rather than paying Alice directly, the latter which could be done by normal public-key cryptography, and of course it doesn't protect against a cheating customer re-publishing the content to the world).

I suspect proxy re-encryption could simplify the design of digital mix schemes like onion routing -- this is left as an exercise for the cryptographically inclinded reader.

This thesis is my source for most of this blog post; it discusses bilinear group cryptography for proxy re-encryption, proxy re-signing, and for reducing the trust needed for blinded offline digital cash.

Legal caveat: many, if not most protocols based on bilinear groups seem to have been recently patented.

Posted by Nick Szabo at 3:31 PM

7 comments:

Anonymous said...

How can one patent math?

6:20 PM

Jeremy Bowers said...

Step one: Apply for a patent on math.

Step two: Accept patent for math from the Patent Office.

It's really pretty simple, Anonymous.

7:25 PM

Francois said...

Well, perhaps that Anonymous is in a country that forbid this kink of patent. For example, in theory, European countries should not allow them.

Blog Archive

- ► 2018 (1)
- ► 2017 (3)
- ► 2016 (4)
- ► 2015 (3)
- ► 2014 (3)
- ► 2013 (3)
- ► 2012 (8)
- ► 2011 (12)
- ► 2010 (9)
- ► 2009 (29)
- ► 2008 (55)
- ▼ 2007 (47)
  - ► December (1)
  - ► October (2)
  - ▼ September (8)
    - Secure time broadcast
    - Real options analysis for space projects
    - Tamper evident numbers
    - Mining the vasty deep (iii)
    - Partial and total orders
    - Institutional changes prerequisite to the industri...
    - Secure property titles
    - Bilinear group cryptography
  - ► August (10)
  - ► July (4)
  - ► June (4)
  - ► May (8)
  - ► April (1)
  - ► March (2)
  - ► February (4)
  - ► January (3)
- ► 2006 (130)
- ► 2005 (44)

12:35 AM

jojo said...

How is this functionally different from simply appending signatures, i.e., Alice signs her document, then Alice's Corp signs her signed document, etc.?

I understand that the transform is a more efficient way to do the above, but functionally, is there any difference? Anything one scheme can accomplish that the other cannot?

9:17 PM

nick said...

See my paper, Elemental Subject Matter for how software (including math) run on computers or stored in memory came to be patentable, even though algorithms and other math are not supposed to be patentable. The basic problem is one of counting gratuitous elements -- elements that do not add anything to the novelty or nonobviousness of a patent, such as a general-purpose memory or CPU, for the purposes of determining whether the subject matter is patentable.

3:44 PM

nick said...

jojo: One basic difference is that when the proxy transforms Alice's signature into Bob's, the proxy does not have to be Bob. Another is that the setup requires cooperation between Alice, Bob, and the proxy during key generation, so that (depending on what out-of-band semantics is imputed to the signature) one can impute some kind of authorization or consent by the parties to the re-signing.

4:01 PM

Anonymous said...

grt

12:35 AM

Post a Comment

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

Simple theme. Powered by Blogger.